

GOOGLE SUMMER OF CODE

LABLUA

INTERRUPT-BASED DRIVERS AND LIBRARIES FOR CEU-ARDUINO

---

## Milestone Report 4

---

*Student:*

Guilherme SIMAS

*Mentor:*

Francisco SANTANNA

June 27, 2017

# Contents

|          |                                     |          |
|----------|-------------------------------------|----------|
| <b>1</b> | <b>Introduction</b>                 | <b>2</b> |
| <b>2</b> | <b>Previous Progress</b>            | <b>2</b> |
| <b>3</b> | <b>Milestone Objective</b>          | <b>2</b> |
| <b>4</b> | <b>Progress</b>                     | <b>2</b> |
| 4.1      | Concurrent Requests . . . . .       | 2        |
| 4.2      | Output Events . . . . .             | 2        |
| <b>5</b> | <b>Conclusion and steps forward</b> | <b>3</b> |

# 1 Introduction

The goal of this report is to show the progress on the project from the past two weeks, comparing it with the expectation from the previous report in order to evaluate the progress made. In the first section the progress from previous report will be summarized. Following the summary, expectations for the proposed two-week period (June 23th) will be listed, followed by what was achieved.

This document concludes by discussing the next steps forward.

## 2 Previous Progress

On the previous report the ADC module was implemented in Ceu as a minimum functioning API. It counts with a function call to request a analog read in a pin and a input event will be emitted by an ISR when the conversion finishes. There were pending issues with the API such as the request not being an output event (as it should be to conform with Ceu's convention) and the undefined behaviour when issuing multiple concurrent requests.

As mentioned in the previous report, a new version of the API whose purpose is to solve these concerns was started but was found to be in an unstable state.

## 3 Milestone Objective

The goal for the current two-week period is to finish the new version which deals with the concerns. As such, requests should be modeled as output events and the routines called in the back-end should somehow treat the concurrent requests.

## 4 Progress

The new API treats the two concerns mentioned above. When developing this version of the API, it was decided that the goal was still to avoid complex implementations that would add to much of a load to the backend or would delay the next modules.

### 4.1 Concurrent Requests

The Atmel datasheet specifies that when a conversion is in progress on the ADC hardware, writing values to the ADSC bit ou ADMUX bits have no effect, since the hardware holds the values. Since the hardware cannot have its registers values changed during a conversion, the `begin()` has virtually no effect when a conversion is not done. Therefore, the `begin()` function is safe.

Another problem remains, which is that the conversion request is lost. To deal with this case, a bit vector is used to keep track of pendant requests. Before calling `begin()`, the driver sets the *i*th least significant bit of the vector, and clears it after emitting the result event. After completing a conversion, the ISR checks if there are conversions left to do (checking the vector's bits). If there are, the driver treats the one with the lowest value (A0 has preference over A1, which has preference over A2 and so on). This means that the highest pins can suffer starvation, but the it was decided that the overhead was not worth it.

### 4.2 Output Events

The Ceu environment provides a file, namely `pinsoutput.c.h`, which lists declarations of output events handlers. In other words, by adding an entry to the file, following the framework, the developer can declare C code to be executed when the user emits an specific output event. This knowledge was gathered by reading through the files and studying how emitting PIN output events worked (behave as a `digitalWrite`).

The idea is for the user to emit an output event in order to issue a requisition to the driver. As such, a new entry in `pinsoutput.c.h` was made for each one of the six analog pins. The code associated

with the output event calls `begin()`, passing in the corresponding analog pin, and adds that pin to the pendent conversions bit vector. As described previously, the `begin()` would be safe since the registers holds the values.

## 5 Conclusion and steps forward

The API fullfils the Ceu convention requirements and is robust enough to be used in applications. Although some improvements remain, such as solving the starvation issue and preventing the user from losing a read result, the cost of implementation is not worth it in terms of project time and computational overhead.

The next two-weeks period will be used to fix any bugs or small tweaks that may be needed for the ADC driver. Furthermore, the focus will be shifting towards the **SPI** module.